BEGINNERS' WORKSHOP

This month, Mike Hardaker unravels the mysteries of ANSI.SYS, while Part Two of Project Menu sees our 'program' take on a prettier face.

Every now and then you come across a program which tells you it needs ANSI.SYS. But what is this, and how do you let the program get at it? First of all, ANSI.SYS is a 'device driver', a special kind of program that alters the way MS-DOS behaves. In order to do this it must be run when the machine starts. This is achieved by including a reference to it in the CONFIG.SYS file (see below).

We looked briefly at CONFIG.SYS in *Beginners Workshop* Issue 53, mentioning that device drivers were most commonly used for handling mice and extra memory. However, device drivers (as their name suggests) can be used to tell MS-DOS about the behaviour of any of the PC's peripheral devices – printers, keyboards, and so on.

ANSI.SYS is a device driver designed to alter the way in which MS-DOS views the 'terminal' attached to the processor – specifically, the display. Its name comes from the acronym for the American National Standards Institute, which was responsible for defining how one of the (many) standard computer terminals worked back in the days when processors filled a room and the terminals were separate items. ANSI.SYS makes the PC behave like an old ANSI terminal.

INSTALLING

To install ANSI.SYS you need to edit the CONFIG.SYS file in the root directory of your system disk (using EDLIN or another editor). First insert the line:

```
DEVICE=ANSI.SYS
```

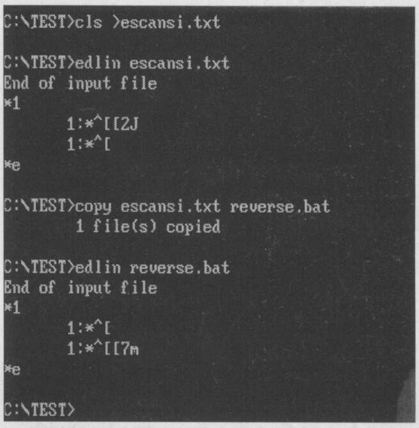
Copy the file ANSI.SYS from your MS-DOS disks into the root directory, reboot (with [Ctrl][Alt][Del]) and you have ANSI.SYS loaded. If you have the file already in a sub-directory on your system disk, you can refer to its location in CONFIG.SYS if you prefer, using a line like:

```
DEVICE=C:\DOS\ANSI.SYS
```

Once it's there, what does it do? In many cases, you won't be aware of it, but ANSI.SYS does allow you rather more control over the display than is standard with MS-DOS and the IBM-compatible PC, particularly if you have a colour monitor. This control extends to defining such things as the colour of the screen (or various parts of it) and whether text is normal, reverse, bold and so on.

ESCAPING

Unfortunately, sending the ANSI-compatible commands to ANSI.SYS is not as straightforward as it might be. This is because ANSI commands are preceded by something called the 'escape' character or ESC. To confuse matters, this has absolutely nothing to do with the [Esc] key on the PC keyboard, but refers to one of the 'invisible' characters that can be used by the PC.



```
C:\TEST>cls >escansi.txt

C:\TEST>edlin escansi.txt
End of input file
*1
       1:*^[[2J
       1:*^[
*e

C:\TEST>copy escansi.txt reverse.bat
        1 file(s) copied

C:\TEST>edlin reverse.bat
End of input file
*1
       1:*^[
       1:*^[[7m
*e

C:\TEST>
```

● The EDLIN ieditor it can be used for simple batch files like REVERSE.BAT (see page 246)

The easiest way of getting an escape character is to redirect the CLS command to a file (as described in last month's *Beginners' Workshop)*. If you enter `CLS > ESCANSI.TXT` and then edit ESCANSI.TXT with your normal editor you will see it contains `^[[2J`

PROJECT MENU – PART TWO

Last month we produced a primitive menuing system and promised that we'd pretty it up a bit this month. Surprisingly enough, we're going to use ANSI.SYS for the job. The first stage is to make the actual Main Menu look snazzier by adding a few colours – if you've got a mono monitor you can't do this bit, although it should be worth your while to read through it.

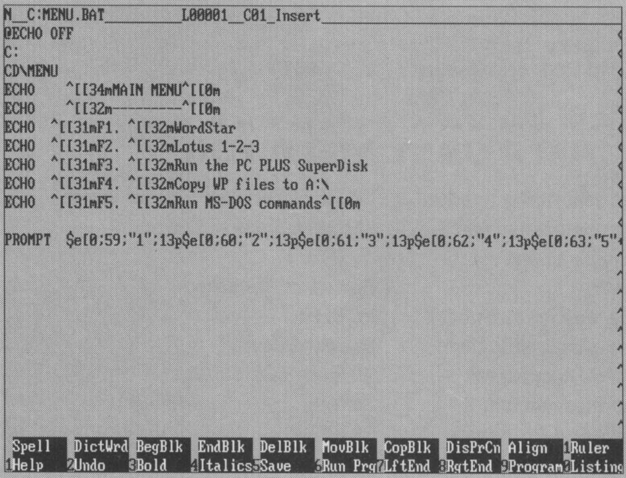
Lets start by making the menu title blue, with a red underline. To do this, change the relevant lines of MENU.BAT so they look like this:

```
ECHO     ^[[34mMAIN MENU^[[0m
ECHO     ^[[31m———-^[[0m
```

Running MENU.BAT should already give you a smarter screen, but it would be nice if the rest of it was in colour too. The next stage is to change the rest of the file so it looks like this:

```
@ECHO OFF
C:
CD\MENU
ECHO    ^[[34mMAIN MENU^[[0m
ECHO    ^[[32m———-^[[0m
ECHO  ^[[31m1. ^[[32mWordStar
```

(continued over page)



```
N__C:MENU.BAT__________L00001__C01_Insert
@ECHO OFF
C:
CD\MENU
ECHO    ^[[34mMAIN MENU^[[0m
ECHO    ^[[32m---------^[[0m
ECHO  ^[[31mF1. ^[[32mWordStar
ECHO  ^[[31mF2. ^[[32mLotus 1-2-3
ECHO  ^[[31mF3. ^[[32mRun the PC PLUS SuperDisk
ECHO  ^[[31mF4. ^[[32mCopy WP files to A:\
ECHO  ^[[31mF5. ^[[32mRun MS-DOS commands^[[0m

PROMPT  $e[0;59;"1";13p$e[0;60;"2";13p$e[0;61;"3";13p$e[0;62;"4";13p$e[0;63;"5"
```

● Working on the new, improved MENU.BAT using the non-document mode of *NewWord*. Most word processors have such a mode (don't try to TYPE the file, by the way – it has the same effect as entering CLS).

The `^[` is the escape character (yes, it looks like two characters, but that's just the way the PC shows it – it isn't the same as typing `^` and then `[`). If you remove the `[2J` part, you now have a file which contains the ANSI escape character and nothing else.

Now, using a text editor or the non-document mode of a word processor (but not EDLIN, which isn't flexible enough) you should be able to insert this character into your batch files by copying, cutting and pasting it. The ANSI commands are at your fingertips.

There are too many of these to list here – look up ANSI.SYS in the index of your MS-DOS manual and you should find a complete list. However, let's look at how we might use one or two of them.

TRYING IT OUT

First of all, we'll create a batch file which sets the screen to inverse video – that is with the background and the text colours swapping places. This batch file is so straightforward that it can even be created using EDLIN.

Start off by loading ANSI.SYS (as described above) and create your ESCANSI.TXT file with only the escape character in it. Then COPY ESCANSI.TXT to REVERSE.BAT. Open this file with EDLIN, list it and you should find you have one line with `^[` on it. Enter '1' at the EDLIN '*' prompt to get the line and add the three characters `[7m` after the escape character. Press enter, then enter [X] to save and exit EDLIN.

Entering **REVERSE** at the prompt should now flip your screen to black text on a white background, in most cases.

To revert to normal, copy REVERSE.BAT to NORMAL.BAT, edit NORMAL.BAT replacing the 7 with 0. As you may have guessed, the '[' and [m] are used to bracket a number which defines the graphics state – with 7 used for reverse and 0 for normal. Your MS-DOS manual will give you a number of other options – try changing your batch file and seeing what happens – but be careful to have NORMAL.BAT always at the ready in case the unexpected occurs.

While graphical commands use '[' and [m], other functions use other characters – we've already seen that `[2J` is used to clear the display when we redirected CLS to a file. To see another function try creating DOWN6.BAT by adding `[6B` to your escape character.

KEEP TRYING

All kinds of fun can be had with ANSI.SYS – just try out different combinations of escape characters and the codes from your MS-DOS manuals and see what happens. Also, look at the examples in the panels of this feature to get some ideas on how best to string the commands together. ●

MORE ESCAPES

There are other ways you can create the ANSI escape code. For example, some word processors allow you to create characters using their built in character number. You do this by holding down the [Alt] key and then typing the relevant number. Although this is a rather laborious way of generating the standard characters, it can be a straightforward way of getting ESC, which is number 027 (the '0' being important). Try it with your word processor but beware – some programs treat the sequence just as if you'd pressed the [Esc] key. If it works, don't bother doing this for every time you need ESC – use a 'dummy' character such as an asterisk while creating the file and then use Search and Replace to swap [Alt]027 for it.

PROMPT FOR AN ANSI

This is an old PC PLUS favourite, but it doesn't hurt to include it again. It uses ANSI.SYS to create a really jazzy prompt in place of the C:\> most of us see. The PROMPT command is actually the easiest way to use ANSI escape codes, as you don't have to figure a way of getting the ESC character into your file – the `$e` parameter for PROMPT does that for you – so `$e[7` will switch your PC screen to reverse video (black text on a white background).

Anyway, try creating this file – see what you think of the new prompt:

```
COPY CON PROMPTER.BAT [Enter]
ECHO OFF [Enter]
(the following two lines must be typed as one)
PROMPT $e[s$e[1;1H$e[K $e[7m $d  $t  $p
$e[0m$e[u$n$g$e[0;59;"dir/p";13p[Enter] [F6]
```

Note that everything from `PROMPT` to `13p` should be entered as one line and that there are three spaces in between each of the `$d`, the `$t`, the `$p` and the `$e`. To run it, just enter PROMPTER at the command line – try pressing [F1] after you've done so.

```
ECHO  ^[[31m2. ^[[32mLotus 1-2-3
ECHO  ^[[31m3. ^[[32mRun the PC PLUS SuperDisk
ECHO  ^[[31m4. ^[[32mCopy WP files to A:\
ECHO  ^[[31m5. ^[[32mRun MS-DOS commands^[[0m
PROMPT $e[47;34m Enter a menu option's number $G $e[0m
```

Now try running the new improved MENU.BAT. If it doesn't seem to be working correctly, check that ANSI.SYS is loaded, check that you've remembered to put in the `[` after the escape character – and remember that the `^[` is the single escape character you created in ESCANSI.TXT.

So far so good – but now lets get even more clever, and use one of the other functions of ANSI.SYS – the ability to control the function keys when combined with PROMPT (this is just as useful for those of you who own mono displays). To control the first menu option with [F1], we need to include the following in the prompt:

```
$e[0;59;"1";13p
```

In this example, the 59 is the code for [F1], with "1" being the command to run 1.BAT (the file loading *WordStar*). The other characters are constants. Therefore, it should be no surprise that `$e[0;60;"2";13p` will drive the second menu option with [F2], `$e[0;61;"3";13p` with [F3] and so on. So change the prompt to look like this:

```
PROMPT
$e[0;59;"1";13p$e[0;60;"2";13p$e[0;61;"3";13p$e[0;62;
"4";13p$e[0;63;"5";13p$e[47;34m Press a Function key $e[0m
```

with the text all on one line. Edit the menu options so they read:

```
ECHO ^[[31mF1. ^[[32mWordStar
```

and so on, and you have a much more elegant menu than when you started off. Next month, we'll make it even better, believe it or not.

HELP US TO HELP YOU

Why not take this opportunity to ask all those niggling questions to which you have been unable to find a satisfactory answer.

There's space here for three but if your list is longer, please write them on a separate sheet. Answers will either be incorporated into future articles or answered on this page.

1 ....

..

2 ....

..

3 ....

..

Please clip out this form (or use a photocopy) and send it to:

Beginners' Workshop,
PC PLUS,
Beauford Court,
30 Monmouth Street,
Bath
BA1 2BW.

Please print clearly:

Name..

Daytime Tel..................................